IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 23, 2017, accepted July 21, 2017, date of publication August 17, 2017, date of current version September 6, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2737544

A Secure Face-Verification Scheme Based on Homomorphic Encryption and Deep Neural Networks

YUKUN MA[1,2], LIFANG WU[1], XIAOFENG GU[1], JIAOYU HE[1], AND ZHOU YANG[1]
[1]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[2]School of Information Engineering, Henan Institute of Science and Technology, Xinxiang 453003, China

Corresponding author: Lifang Wu (lfwu@bjut.edu.cn)

ABSTRACT With the increase in applications of face verification, increasing attention has been paid to their accuracy and security. To ensure both the accuracy and safety of these systems, this paper proposes an encrypted face-verification system. In this paper, face features are extracted using deep neural networks and then encrypted with the Paillier algorithm and saved in a data set. The framework of the whole system involves three parties: the client, data server, and verification server. The data server saves the encrypted user features and user ID, the verification server performs verification, and the client is responsible for collecting a requester's information and sending it to the servers. The information is transmitted among parties as cipher text, which means that no parties know the private keys except for the verification server. The proposed scheme is tested with two deep convolutional neural networks architectures on the labeled faces in the Wild and Faces94 data sets. The extensive experimental results, including results for identification and verification tasks, show that our approach can enhance the security of a recognition system with little decrease in accuracy. Therefore, the proposed system is efficient with respect to both the security and high verification accuracy.

INDEX TERMS Face verification, Paillier encryption, convolutional neural network.

I. INTRODUCTION

Secure Internet identity authentication is essential for Internet applications. Nowadays, frequent network security accidents reveal the disadvantages of traditional identity authentication based on user IDs and passwords. The safety of the traditional system cannot be ensured because hackers can intrude on a system if they have obtained the ID and password. More importantly, this verification method separates the digital identity from the physical identity, which is also very inconvenient.

To address the disadvantages of traditional identity authentication, there has been a rapid development in biometric identification technology in recent years. Human biometrics includes physical and behavior characteristics. A physical characteristic is a feature that the user has, such as a fingerprint, iris, or face. A behavior characteristic is the manner in which a user performs an action, such as gait, keystroke, or signature. Of all of these biometrics, the face has unique advantages as a verification method with good results obtained using deep learning.

However, a biometric template affects a user's privacy, and furthermore, it is irrevocable. Therefore, ensuring the safety of a user's privacy is a precondition for the applications of biometrics authentication systems. Not only should a good biometric verification system utilize biometric features to verify identity with sufficient accuracy, it must also ensure the features' variety, tractability, and security [1]. In recent years, many breakthroughs have occurred. In 2010, Osadchy developed a system called SCiFI (Secure Computation of Face Identification) [2], in which the identification was accomplished in a secure way that could protect both the privacy of the subjects and the confidentiality of the database. Every face image is represented by a binary vector using a local image patch-based method. Later, Luong proposed a method to attack the SCiFI system by reconstructing a fragmented face [3]. To make up for this disadvantage, in 2015, Jin et al. used sparse representation instead of p patches to prevent the patch-based attack of SCiFI system [4].

However, two major drawbacks still exist in the systems above: (1) The client knows the privacy keys for decryption,

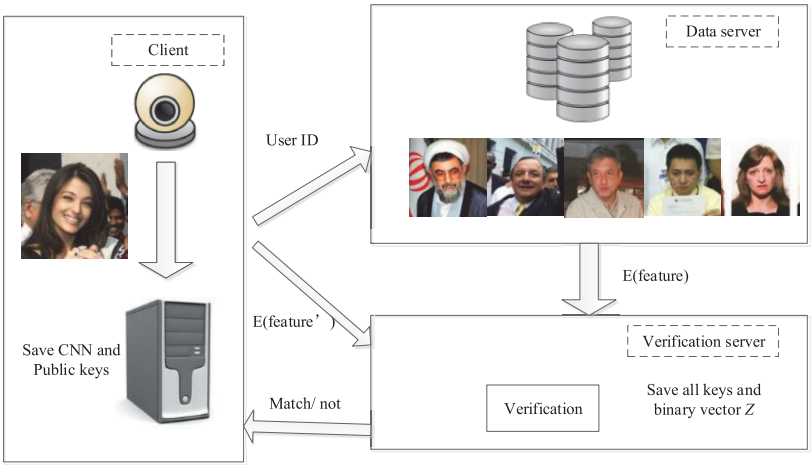


FIGURE 1. Authentication system.

and as clients are distributed at every location at which the application is used, potential threats can influence the security of the whole system. (2) The accuracies are relatively low, in spite of the large data volume of the face representation.

To ensure the face features' variety, tractability, and security, in this paper, we propose a novel face verification system. In this system, a convolutional neural network (CNN) is used to extract features and binarize them as face representations. The features are then encrypted with the Paillier algorithm and saved in a dataset. The framework involves three parties: the client, data server, and verification server, as shown in Figure 1. The user features are saved in the data server in cipher text and all the information is transmitted as cipher text, which prevents a user's features from being stolen or maliciously distorted, and the client is only involved in the final verification result. Furthermore, the server is divided into two parts: data server and verification server; thus, the function of each part is clearer.

The rest of paper is organized as follows. Section II of this paper introduces the proposed scheme, including the feature extraction and homomorphic encryption. Section III presents the secure authentication scheme, and Section IV presents the experiment and discusses the proposed system. Finally, the paper is concluded in Section V.

II. PROPOSED SCHEME

A. FACE REPRESENTATION

In early research, most of the features in face recognition algorithms were fixed and handcrafted, such as LBPs, SIFTs, and Gabor features. In recent years, face features based on CNNs have been proven to be more efficient for face verification [5], [6]. In a CNN, the neurons of the different layers indicate different input information, and the last-hidden layer is usually used as the face representation. Suppose the number of neurons in the last-hidden layer is n, and this layer is extracted as the face feature. To remove noise for later encryption, the outputs are binarized, i.e., any stimulation values greater than 0 are set to 1, as shown in Figure 2.

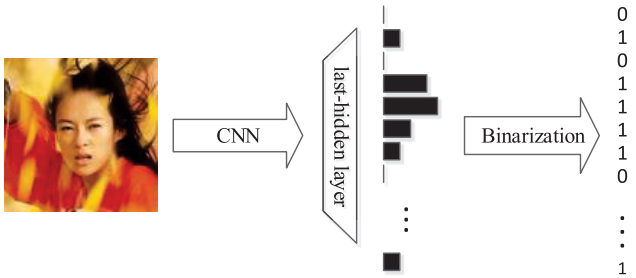


FIGURE 2. Face representation.

B. PAILLIER CRYPTOSYSTEM

Although verification based on a binary face representation is feasible, saving the face features directly in the system is unsafe. Zeiler *et al.* proposed a method to visualize the input image from the neuron stimulation value, and obtained most of the pixel-level information [7], which should be protected to preserve the user's privacy. Furthermore, criminals who obtain the face features from one system can easily invade other systems. Thus, features must be encrypted to ensure their safety before they are loaded into the system.

We encrypt the face features using the Paillier encryption algorithm, which is an asymmetric algorithm [8]. This means that the keys are divided into public keys (for encryption) and privacy keys (for decryption). Here, we mainly analyze the homomorphism of the Paillier encryption.

The Paillier scheme is additive homomorphic, which enables it to compute $E(m_1 + m_2)$ given two encryptions $E(m_1)$ and $E(m_2)$ without knowledge of the private key. It can also compute $E(c \cdot m_1)$ for any known constant c. The detailed algorithms are as follows:

$$D\left(E(m_1, r_1) \cdot E(m_2, r_2) \bmod n^2\right) = m_1 + m_2 \bmod n \quad (1)$$

$$D\left(E(m_1, r_1)^{m_2} \bmod n^2\right) = m_1 m_2 \bmod n \quad (2)$$

where $D(x)$ is the decrypted result of cipher text x, $E(m, r)$ is the cipher text of m based on random number r, and n is a

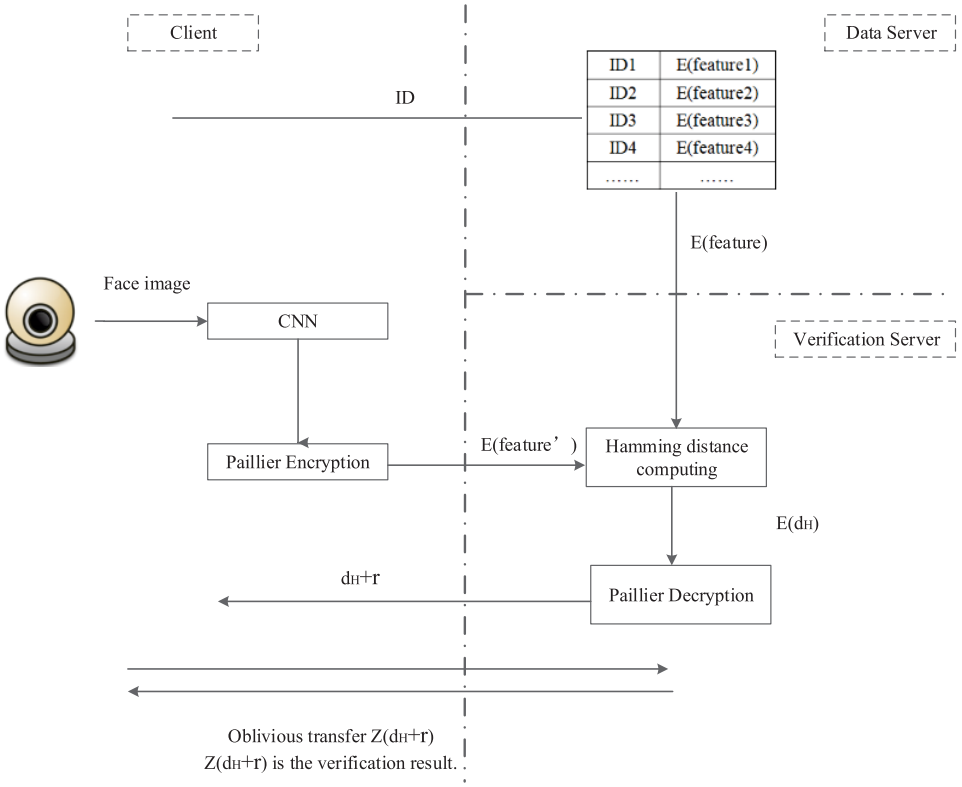


FIGURE 3. Verification step.

parameter needed for key generation. Parameter n should be set to the product of two large prime numbers.

C. DISTANCE CALCULATION

Consider two binary numbers m_1 and m_2, and let e be their exclusive-or value. Then, e can be expressed as

$$\begin{aligned} e = (m_1 - m_2)^2 &= {m_1}^2 + {m_2}^2 - 2m_1m_2 \\ &= m_1 + m_2 - 2m_1m_2 \end{aligned} \tag{3}$$

An additive and multiplicative homomorphism can then be obtained as follows.

$$D\left(E(m_1)E(m_2)/[E(m_1)]^{2m_2} \bmod n^2\right) = e \tag{4}$$

The Hamming distance between two binary vectors can be evaluated by $D\left(\prod_i E\left(e_i\right) \bmod n^2\right)$, where e_i is the exclusive-or value of the ith bit.

All of the above prove that, if $\mathrm{E}\{X\} = \{\mathrm{E}[x_1], \mathrm{E}[x_2], \cdots, \mathrm{E}[x_n]\}$, $\mathrm{E}\{Y\} = \{\mathrm{E}[y_1], \mathrm{E}[y_2], \cdots, \mathrm{E}[y_n]\}$, and $Y = \{y_1, y_2, \cdots, y_n\}$ are known, where $X = \{x_1, x_2, \cdots, x_n\}$, the server can compute the cyphertext of the Hamming distance between X and Y without decrypting $E\{X\}$.

III. SECURE VERIFICATION SYSTEM

The verification system includes three parties: the client, data server, and verification server, which are separated by dotted and dashed lines in Figure 3. The client saves the public keys of the Paillier algorithm, and the verification server saves the public and private keys.

Face verification in this paper is based on the Hamming distance between two face features. When the Hamming distance is greater than a threshold (chosen ahead of time), the faces are regarded as belonging to different individuals, which means the verification fails. For security, the verification server also saves an additional binary vector $Z = \{z_0, z_1, z_2, \cdots, z_k\}$, where k is the length of the face feature. The definition of z_i is

$$z_i = \begin{cases} 0 & \text{when } i < d_{threshold} \\ 1 & \text{when } i \geq d_{threshold} \end{cases} \tag{5}$$

where $d_{threshold}$ is the Hamming distance threshold for verification.

In the registration stage, the client obtains a face image and extracts the face feature using CNN. It encrypts the feature using the public keys of the Paillier algorithm. Finally, the client sends the cipher text of the face feature and user ID to the data server to complete the registration.

Details of the verification stage are illustrated in Figure 3. The steps are as follows:

1) The client extracts the feature from a user's face image and encrypts it.

2) The client sends the user ID to the data server and sends the cipher-text of the feature $E(feature')$ to the verification server (the notation $feature'$ is used to distinguish it from $feature$ during registration).

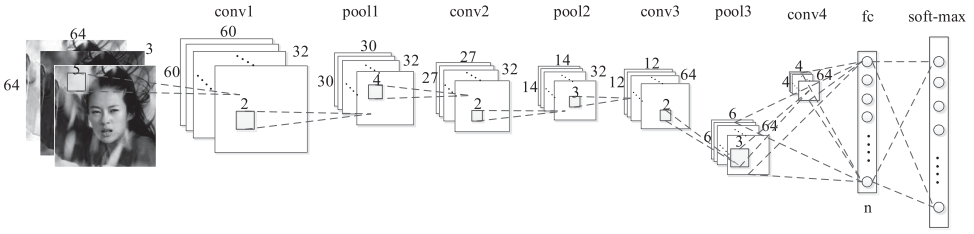


FIGURE 4. DeepID network.

3) The data server searches for the feature cipher-text $E(feature)$ that corresponds to the user's ID and sends it to the verification server.

4) The verification server computes the cipher text of Hamming distance $E(d_H)$ based on Paillier encryption homomorphism and decrypts it to d_H. The calculation of this method is described in Section 2.3 .

5) To prevent the client from obtaining the value of d_H directly or allowing it to be distorted during the transmission, the verification server chooses a random number r and sends $d_H + r$ to the client. At the same time, the verification server computes $X = \{x_0, x_1, x_2, \cdots, x_k\}$, and $x_i = z_{i-r \bmod k}$, i.e., X is the result of circular bit-shifting Z by r places.

6) Invoking of one of k oblivious transfer protocols OT_1^k (explained below), the client and verification server map $d_H + r$ to the appropriate output value x_{d_H+r}, namely z_{d_H}. In this step, the client is the receiver and the verification server is the sender.

In an OT_1^k protocol, a sender transfers one of many pieces of information to a receiver, but remains oblivious as to which piece has been transferred [9]. Suppose that A has several secrets and sends one of them to B, but only B knows which secret A has transferred to it. Suppose A's secrets are $\{s_1, s_2, s_3, \cdots, s_k\}$, where s_i is binary value. The protocol is as follows:

Party A gives a one-way function f to B and keeps f^{-1} secret. If B needs s_i, then B randomly chooses k numbers $x_1, x_2, x_3, \cdots, x_k$ and sends $y = \{y_1, y_2, y_3, \cdots, y_k\}$ to A, where y_j is

$$y_j = \begin{cases} x_j & j \neq i \\ f(x_j) & j = i \end{cases} \tag{6}$$

Party A then computes $z_j = f^{-1}\left(y_j\right) (j = 1, 2, \cdots k)$ and sends $z_j + s_j (j = 1, 2, \cdots k)$ to B. As $z_i = f^{-1}(y_i) = f^{-1}(f(x_i)) = x_i$, , B can obtain s_i from $z_j + s_j$ using x_i without knowing anything about $s_j (j \neq i)$.

IV. EXPERIMENTS AND SYSTEM EVALUATION

A. SETUP

We tested the proposed face verification scheme using deep convolutional neural network architectures on the Labeled Faces in the Wild (LFW) and Faces94 datasets. We consider the following deep neural network architectures:

TABLE 1. Architecture of the Light CNN-9 model.

Type	Filter Size /Stride, Pad	Output Size	Parameters
Conv1	5×5/1, 2	128×128×96	2.4K
MFM1	-	128×128×48	-
Pool1	2×2/2	64×64×48	-
Conv2a	1×1/1	64×64×96	4.6K
MFM2a	-	64×64×48	-
Conv2b	3×3/1,1	64×64×192	82.9K
MFM2b	-	64×64×96	-
Pool2	2×2/2	32×32×96	-
Conv3a	1×1/1	32×32×192	18.4K
MFM3a	-	32×32×96	-
Conv3b	3×3/1,1	32×32×384	331.8K
MFM3b	-	32×32×192	-
Pool3	2×2/2	16×16×192	-
Conv4a	1×1/1	16×16×384	73.7K
MFM4a	-	16×16×192	-
Conv4b	3×3/1,1	16×16×256	442.4K
MFM4b	-	16×16×128	-
Conv5a	1×1/1	16×16×256	32.8K
MFM5a	-	16×16×128	-
Conv5b	3×3/1,1	16×16×256	294.9K
MFM5b	-	16×16×128	-
Pool4	2×2/2	8×8×128	-
fc1	-	512	4194.3K
MFM-fc1	-	256	-

1) DeepID [5]

We use this network with small modifications, as shown in Figure 4. The network contains four convolutional layers, followed by the fully-connected layer and softmax output layer. In Figure 4, the numbers alongside the rectangles denote the number of feature maps, while the width and height denote the dimensions of the feature map. The fully connected layer is extracted as the face feature.

2) LIGHT CNN-9 [10]

The details of the Light CNN-9 model are presented in Table 1. Instead of ReLU (Rectified Linear Units), the MFM layer is used as activation function. Here, x^n $(n = \{1, 2, \cdots, 2N\})$ is an input convolution layer and the MFM 2/1 operation is

$$\widehat{x}_{i,j}^k = \max\left(x_{i,j}^k, x_{ij}^{k+N}\right) \tag{7}$$

The MFM 2/1 operation combines $2N$ feature maps into N feature maps with the same width and height. Compared to ReLU, the MFM function makes CNN models light and robust [10]. in this model, the MFM-fc1 layer is used as the face feature.

In our experiments, we used cuda-convnet [11] and caffe [12], which are popular DNN toolboxes. We used the

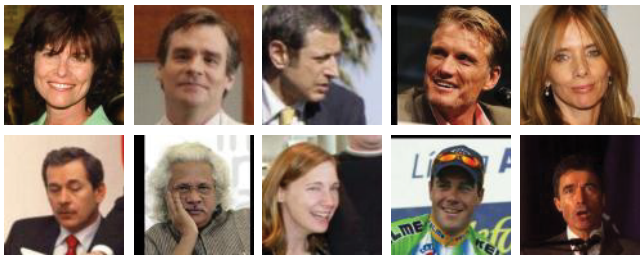


FIGURE 5. Example images from the training and testing datasets: (top row) CASIA-WebFace and (bottom row) LFW.

FIGURE 6. Example of data augmentation images.

TABLE 2. Accuracy comparison with/without encryption.

Networks	LFW (%)	Faces94 (%)
DeepID(No encryption)	93.6	96.24
DeepID(Encryption)	93.27	94.53
Light CNN-9(No encryption)	98.11	99.0
Light CNN-9(Encryption)	95.73	96.75

CASIA-WebFace database [13] to train the CNN and the LFW database [14] for verification. Both datasets consist of unconstrained face images and include variation in pose, facial expression, illumination, and occlusion. Figure 5 shows several example images. The LFW data set contains 13,233 images of 5,749 people collected from the web. Only 1,680 of the people have two or more distinct photos in the data set, so it is not suitable for training. The CASIA-WebFace dataset contains 10,575 subjects and 494,414 images. The identities in CASIA-WebFace do not intersect those in LFW. We trained the CNN using the identification information for supervision. We used 90% of the images for each individual as the training set, while took the remaining 10% of the images to test the CNN system and train the verification classifier. In order to increase the size of the training set and enhance the robustness of the system, we performed data augmentation on the training set. Specifically, we horizontally flipped and applied color filters to the original image, as shown in Figure 6; hence, the training set is increased three times.

B. VERIFICATION ACCURACY

After training the deep neural network with the CASIA-WebFace training set, we extracted the face features of LFW and encrypted them based on Paillier encryption. Then, face verification was performed based on the Hamming distance, which is computed in cypher text.

For comparison, we also conducted the verification experiment on the Faces94 database [15]. This database includes 20 pictures of 152 individuals and consists of 3,040 pictures in total.

FIGURE 7. Examples of verification failures.

Experimental results are given in Table 2, where we can see that our proposed scheme obtained the security of system at the expense a slightly decrease in accuracy. Figure 7 shows some failure verification examples of LFW. Most are because of occlusion, exaggerated expressions, or similarities among different people.

For further validation of the proposed scheme, we also performed face identification on the Faces94 dataset. Using the same experimental setting in [4], we chose five registered face pictures for each individual (100 images total), then selected 100 pictures randomly from the rest for identification. We compare the accuracy of the top-1 match and feature volume with other methods in Table 3. The proposed scheme can obtain highest accuracy with smallest feature volume.

TABLE 3. Comparison with other methods for identification task.

Method	[4]	[2]	ours
Accuracy of the top-1 match	91.55%	95.5%	96.6%
Feature volume	1,600 bits	3,000 bits	256 bits

C. ACCELERATION

We tested the computing time of each main step. The PC on which the experiment was conducted was equipped with Ubuntu 14.04, 32 GB memory, an i7-5930K CPU, and a GeGorce GTX980 GPU.

The first step, feature extraction based on the DeepID network, takes 30 min to process all 13,233 pictures of LFW, which is 0.14 s per picture on average. The second step, feature encryption, takes 0.57 s to encrypt a 256-dimensional binary feature using Paillier encryption. The third step, calculating the Hamming distance on the verification server in the cipher domain, is 1.9 ms, while it takes 0.01 ms to calculate the plaintext Hamming distance. In the experiment, we implemented the three parties on the same PC, so we did not take the transmission delay into consideration.

From the result above, we can see that most of the time is spend on feature encryption. Some optimization can be implemented to shorten this time. As all the features to be encrypted have two possible values (0 or 1), and the client could generate several cipher texts of 0s and 1s in advance when the system is idle (i.e., there are no verification requests). When a verification request arrives, the client

merely needs to use the cipher text corresponding to the feature values. It would shorten the time of encryption as much as possible and ensure the speed of the system.

D. DATA VOLUME EVALUATION

Using Paillier encryption on the face representations will inevitably increase the data volume saved on the data server. We hence quantitatively analyze the data size before and after encryption. Before encryption, the feature has two values, so it is 1 bit per dimension. After encryption, the range of values of the cipher text is $\left[0, n^2\right)$, $n = p \cdot q$ (where p and q are large prime numbers selected in the Paillier algorithm). Therefore, saving a one-dimensional feature requires $\log_2 n^2$ bits.

E. SECURITY EVALUATION

In the encryption algorithm, only the plaintext and keys should be kept secret. Hence, we should assume that the attacker knows the encryption algorithm and cipher text. Therefore, the security of the encryption relies on the secret key rather than the ignorance of the hackers about the encryption algorithm. Paillier encryption can provide semantic security against chosen plaintext attacks (CPA), so the cipher text in this paper is CPA-safe.

The security of Paillier encryption is equal to the decisional composite residuosity assumption, which states that given a composite n and integer z, it is hard to decide whether z is an n-residue modulo n^2 or not. The difficulty of decrypting the cipher text is hence equal to the difficulty of determining composite residuosity.

In the overall system, all the information is transferred as cipher text to prevent attackers from stealing or distorting the information. The client only knows the results of the verification. Because the clients cannot obtain the Hamming distance value, they cannot conduct a brute-force attack by changing the input images. Features in both the data server and verification server are shown as cipher text, protecting the plaintext feature from being stolen and revealing its pixel-level information.

V. CONCLUSION

In this paper, we propose a secure face-verification system based on a CNN representation. In this system, all face features are saved in cipher text, and the client know only the verification result (whether the face matches or not). Under the premise of ensuring security, our system achieves more accurate verification than comparable methods and high efficiency, which satisfies the requirement for real-time operation.

REFERENCES

[1] A. B. J. Teoh, D. C. L. Ngo, and A. Goh, "Personalised cryptographic key generation based on FaceHashing," *Comput. Secur.*, vol. 23, no. 7, pp. 606–614, 2004.

[2] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "SCiFI—A system for secure face identification," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, May 2010, pp. 239–254.

[3] A. Luong, M. Gerbush, B. Waters, and K. Grauman, "Reconstructing a fragmented face from a cryptographic identification protocol," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2013, pp. 238–245.

[4] X. Jin *et al.*, "Privacy preserving face identification in the cloud through sparse representation," in *Proc. Chin. Conf. Biometric Recognit.*, Cham, Switzerland, 2015, pp. 160–167.

[5] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. CVPR*, Jun. 2014, pp. 1891–1898.

[6] Y. Sun, X. Wang, and X. Tang. (Jun. 2014). "Deep learning face representation by joint identification-verification." [Online]. Available: https://arxiv.org/abs/1406.4773

[7] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland, 2014, pp. 818–833.

[8] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Berlin, Germany, 1999, pp. 223–238.

[9] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," in *Proc. Annu. Int. Cryptol. Conf.*, Berlin, Germany, 1999, pp. 573–590.

[10] X. Wu *et al.* (2015). "A light CNN for deep face representation with noisy labels." [Online]. Available: https://arXiv preprint arXiv:1511.02683

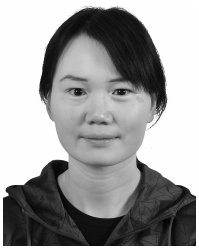
[11] (Jul. 2012). A. Krizhevsky. *Cuda-Convnet*. [Online]. Available: http://code.google.com/p/cuda-convnet/

[12] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia (MM)*, 2014, pp. 675–678.

[13] D. Yi, Z. Lei, S. Liao, and S. Z. Li. (Nov. 2014). "Learning face representation from scratch." [Online]. Available: https://arxiv.org/abs/1411.7923

[14] G. B. Huang *et al.*, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, 2007.

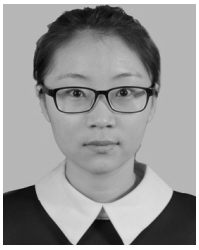
[15] L. Spacek, "Collection of facial images: Faces," in *Computer Vision Science and Research Projects*, vol. 94. Colchester, U.K.: University of Essex, 2007. [Online]. Available: http://cswww.essex.ac.uk/mv/allfaces/faces94.html



YUKUN MA is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. Her research interests include face recognition and biometric encryption.

LIFANG WU received the bachelor's, master's, and Ph.D. degrees from Beijing University of Technology, Beijing, China, in 1991, 1994, and 2003, respectively. She is currently a Professor with Beijing University of Technology. Her research interests include social recommendation, face encryption, and deep learning-based video analysis.

XIAOFENG GU was born in 1992. He received the bachelor's degree from Beijing University of Technology, Beijing, China, in 2016. His research interests include image processing, deep learning, and biometric encryption.



JIAOYU HE was born in 1993. She is currently pursuing the M.S. degree with Beijing University of Technology, Beijing, China. Her research interests include image and video processing and deep learning.

ZHOU YANG was born in 1994. He is currently pursuing the bachelor's degree with Beijing University of Technology, Beijing, China. His research interests include image processing.